NT DII COE Application Installation Procedure

Proposal

23 April 1998

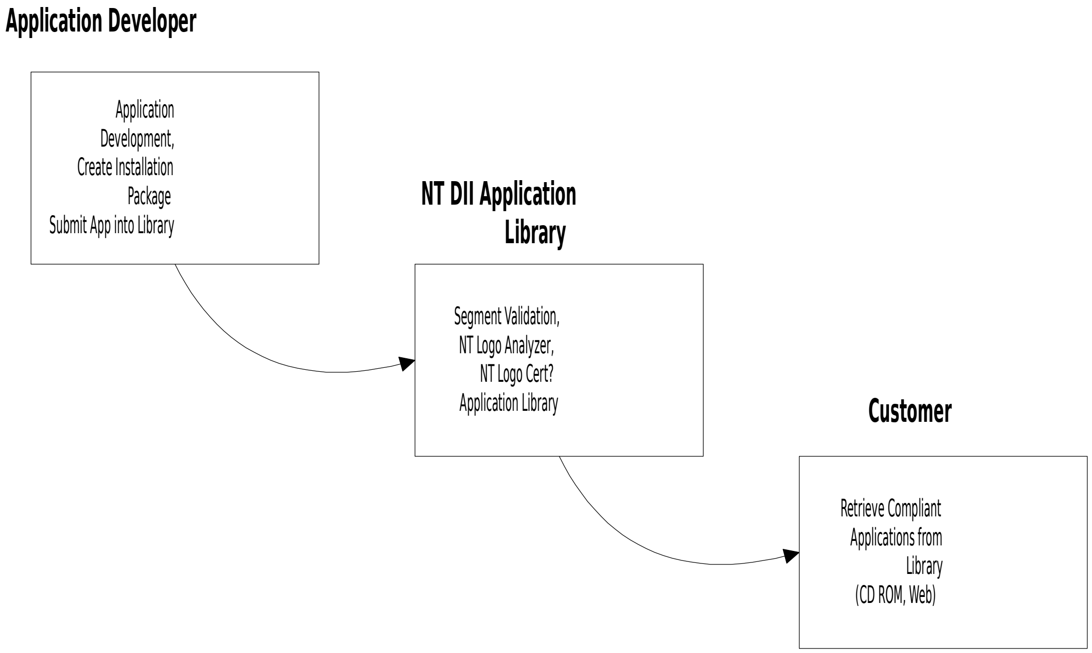
Presented by
Andrew Cox
PMW 171-1

MITRE

Agenda

- Introduction
- Objectives
- NT Logo and I&RTS Compliance
- Criteria for Selecting COTS Products
- Design Approach
  - » Registry Issues
  - » Why XML
- Sample XML Application Description

NT Development Cycle



Application Developer

Application Development,
Create Installation Package
Submit App into Library

NT DII Application Library

Segment Validation,
NT Logo Analyzer,
NT Logo Cert?
Application Library

Customer

Retrieve Compliant Applications from Library
(CD ROM, Web)

Objectives

- Create a Standardized Application Installation and Validation process for use by Application Developers.
- Maximize use of COTS technologies
- Adhere to NT Logo installation Requirements
  - » Automate the Creation of Installation Package
- Innovative Approach to satisfying I&RTS Tenants
  - » Goes beyond NT logo Rqmts - Federation of Software

Criteria for Selecting COTS Products

- Runs on Windows 95 and NT
- 32-bit Graphical User Interface
- Programmable
- Flexible for specific installation needs
- Windows Logo Compliant Installation

NT Logo Compliance Installation Rqmts

- Be Easy to Install and Remove
- Provide Graphical 32-bit setup
- Provide for Unattended Installation
- Support Installation from multiple CD-ROMs
- Support Add/Remove Programs
- Avoid Rebooting the System
- Detect and Handle Versions
- Register an Uninstaller

NT Logo Compliance Uninstallation Rqmts

- Remove All Application Files during Uninstall
- Remove all References from the Start Menu
- Remove Registry Entries during Uninstall
- Remove Uninstaller as the Last Step of Uninstall

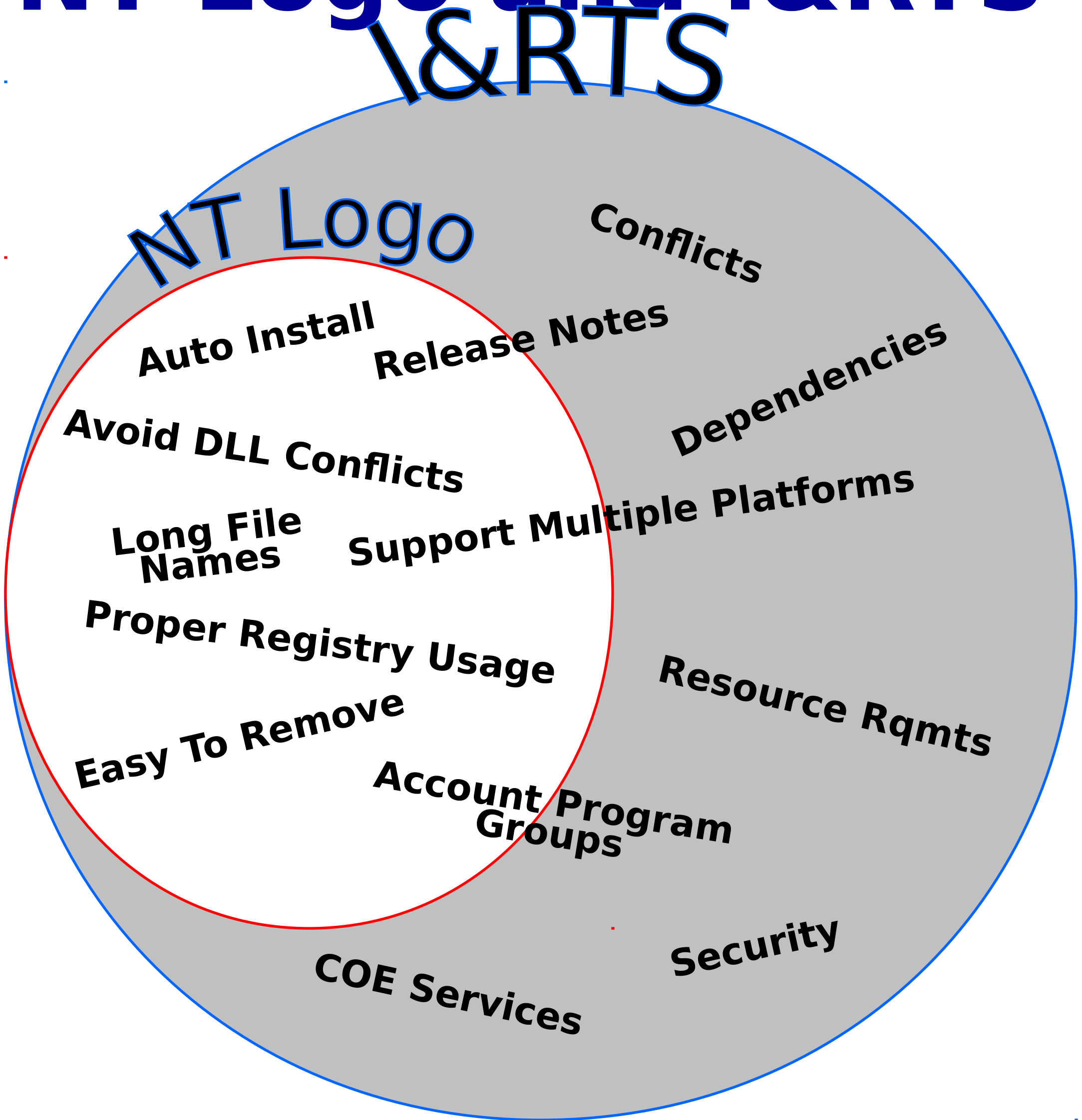
Recommend COTS InstallShield 5.1

- Supports NT Logo Compliant Installation
- Programmable (C like scripting language)
- Allows Execution of 3rd party applications within InstallShield
- Supplies and Registers an Automated Uninstaller
- Logs all of its Actions
- Supports Interactive and Silent (hands-off) Installation
- Checks for User Privilege Level and Installs Accordingly

Recommended Application Descriptor

- Must Support the DII COE I&RTS Rqmts
  - » Conflicts, Software/File Dependencies
  - » System Requirements
  - » Shared Files, File Attributes, Permissions
  - » COE Services
  - » Version Control
- Must Support NT Logo Compliance
  - » Developer Information, Versions, Release Notes, Help
  - » Efficient Use of Registry

Relationship Between NT Logo and I&RTS



What Should Go in the Registry?

Source:
Inside Windows 95 Registry
by Ron Petrusha

- Application Settings
  - » NT Logo Compliant Information
  - » Installation and Licensing Information
- User Settings
  - » Configurable Options: Tool Bars, Default Directories, Colors, etc
- Sound Events
  - » Program Errors, Closing Windows, Warnings, Messages, etc
- Application State Information (position)
- Most Recently Used (MRU) Files List

Registry Storage Space

Source:
January 1998 MSDN

- Generally, data consisting of more than one or two kilobytes (K) should be stored as a file and referred to by using a key in the registry
- Each Self Registering File Creates 50 Registry Entries
- Limit 55,000 Entries

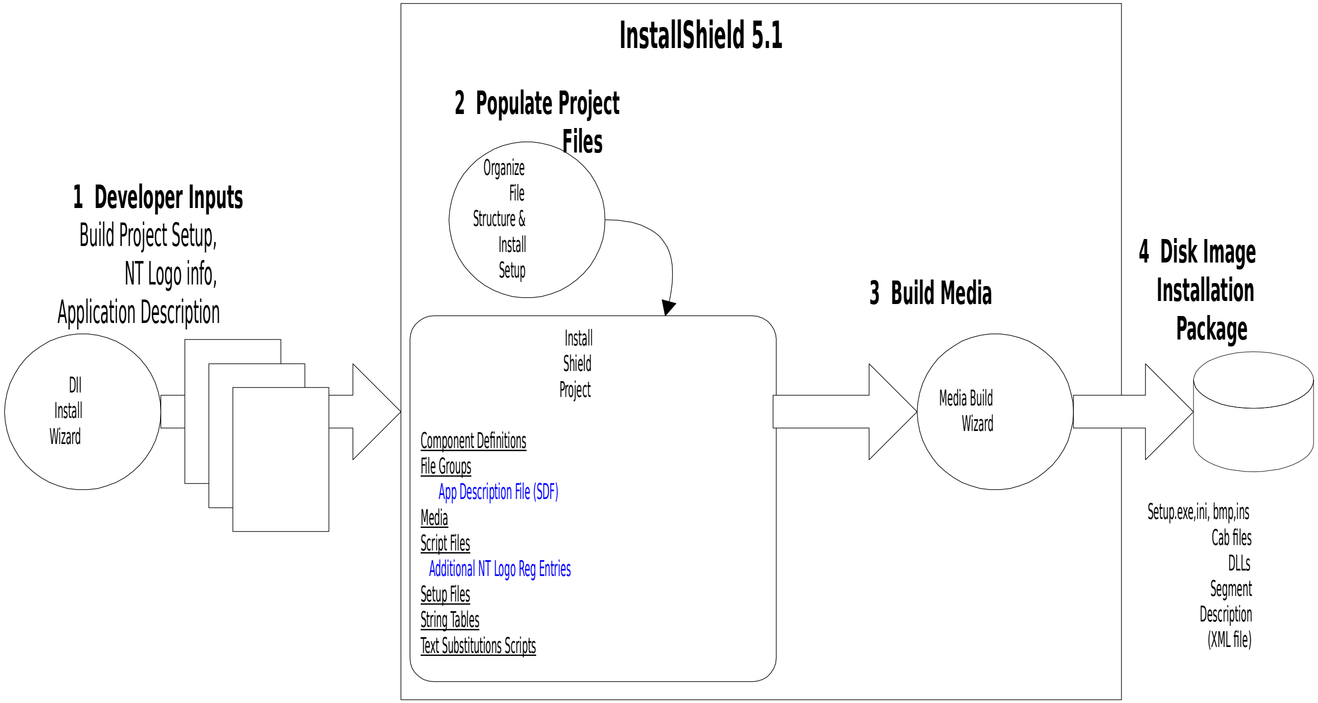
What Should NOT go in the Registry

- Crime
  - » Saving System Info That Is Available Via Windows API
  - » Saving Non-Application State Like Information
  - » Exceeding Max Configured Registry Size
- Punishment
  - » Performance Degradation
  - » Error: Cannot Write to Registry (serious)
  - » Error: Corrupt Registry (deadly, must re-install Windows)
  - » Non-Compliance with Future Limitation on Registry Size

Recommended Application Descriptor File Format

- Extensible Markup Language (XML)
- Human Readable
- Supports User Defined Tags (Document Type Definition)
- Free Parsers Available
- Major Browser Developers Pledged Support of XML
- Language Independent

Installation Package Development Process



InstallShield 5.1

2 Populate Project Files

Organize File Structure & Install Setup

1 Developer Inputs

Build Project Setup,
NT Logo info,
Application Description

DII Install Wizard

Install Shield Project

Component Definitions
File Groups
App Description File (SDF)
Media
Script Files
Additional NT Logo Reg Entries
Setup Files
String Tables
Text Substitutions Scripts

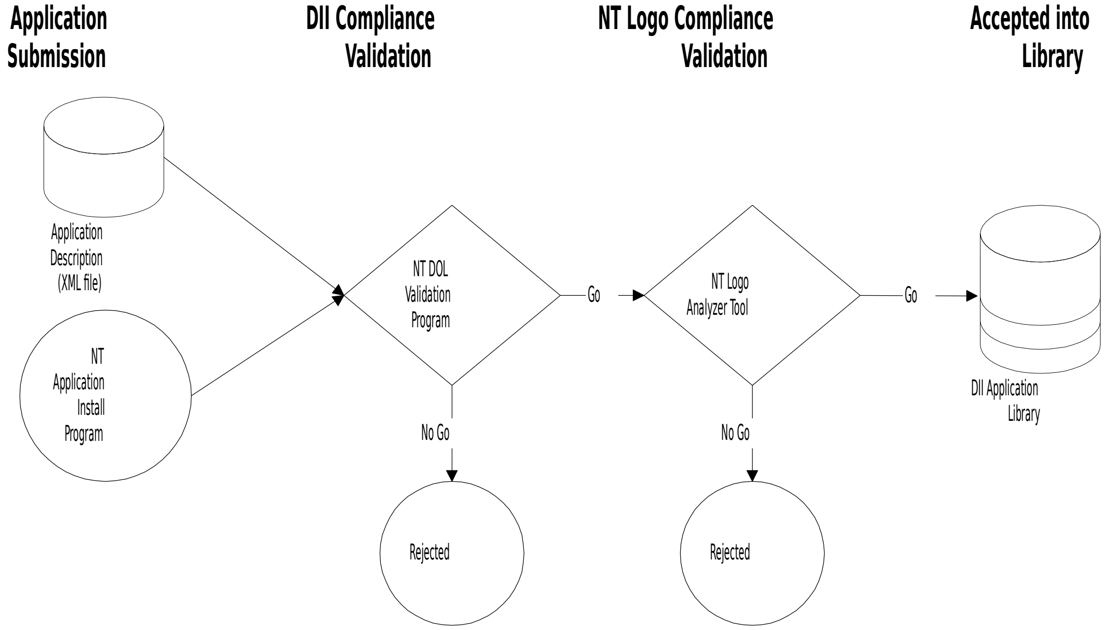
3 Build Media

Media Build Wizard

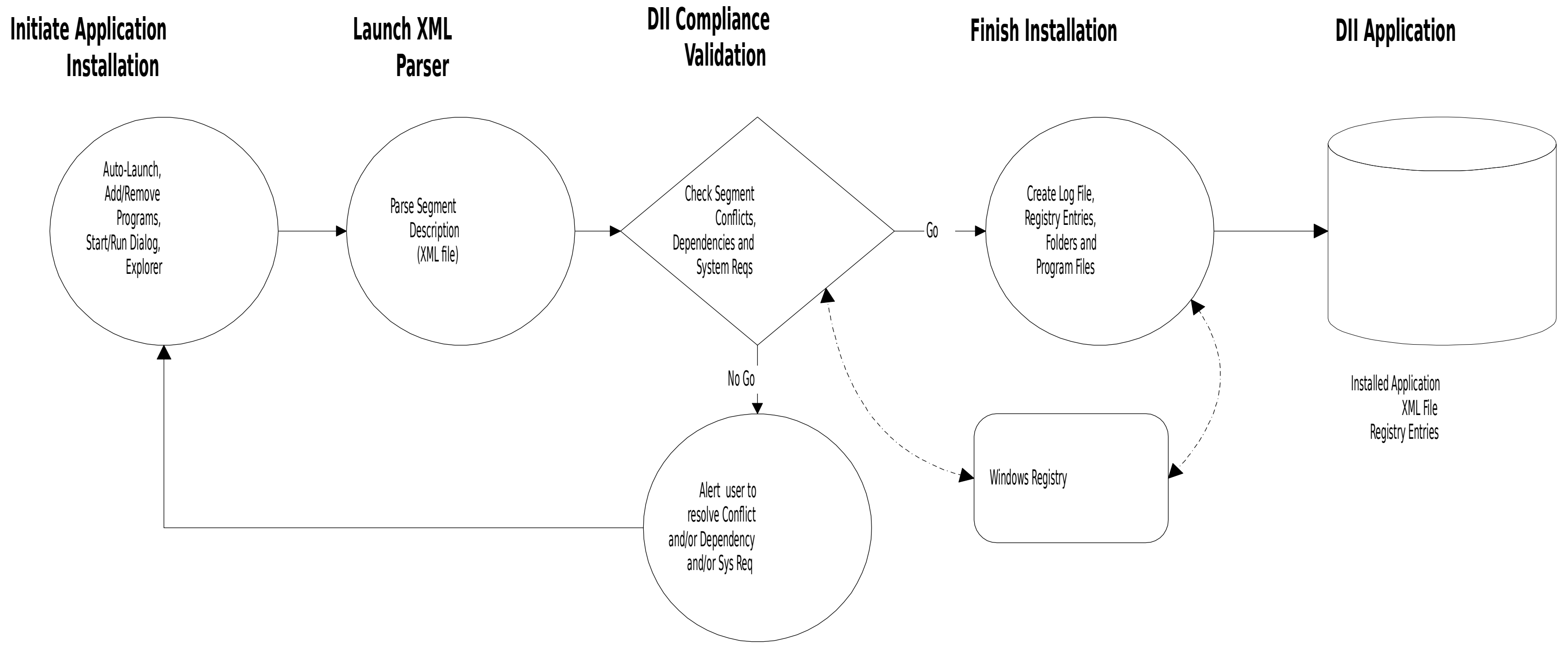
4 Disk Image Installation Package

Setup.exe,ini, bmp,ins
Cab files
DLLs
Segment Description (XML file)

NT DII Application Library


Application Submission
DII Compliance Validation
NT Logo Compliance Validation
Accepted into Library
Application Description (XML file)
NT Application Install Program
NT DOL Validation Program
Go
No Go
Rejected
NT Logo Analyzer Tool
Go
No Go
Rejected
DII Application Library

Application Installation Process


Initiate Application Installation
Launch XML Parser
DII Compliance Validation
Finish Installation
DII Application
Auto-Launch, Add/Remove Programs, Start/Run Dialog, Explorer
Parse Segment Description (XML file)
Check Segment Conflicts, Dependencies and System Reqs
Go
Create Log File, Registry Entries, Folders and Program Files
Installed Application XML File Registry Entries
No Go
Alert user to resolve Conflict and/or Dependency and/or Sys Req
Windows Registry

Sample Application Description XML File

```
<?XML VERSION="1.0"?>
<!DOCTYPE SegmentDesc SYSTEM "SegmentInfo.dtd">
<SegmentDesc>

<!--

*********************************************************************
    ** The Name of the Segment with It's classification required.
    ** The Id element is the Unique Id of the Segment in the DOL
Database

*********************************************************************
-->
    <Name Id="Sample-24" Classification="UNCLASS">Sample Segment</Name>


<!--

*********************************************************************
    ** The Version number of the Segment

*********************************************************************
-->
    <VersionNumber>3,1,0,4</VersionNumber>

<!--
```

Sample Document Type Definition (DTD)

```
<!ELEMENT Name (#PCDATA)>
<!--

*********************************************************************
    ** The Name of the Segment with It's classification required.
    ** The Id element is the Unique Id of the Segment in the DOL
Database

*********************************************************************
-->
<!ATTLIST Name
    Id             ID  #REQUIRED
    Classification (UNCLASS|CONFIDENTIAL|SECRET|TOPSECRET) "UNCLASS">

<!--

*********************************************************************
    ** A name/version structure used to track
    ** required segments and conflicts

*********************************************************************
 -->
<!ELEMENT VersionNumber (#PCDATA)>
<!ELEMENT SegmentName (#PCDATA)>
<!ELEMENT SegmentVersion (SegmentName, VersionNumber)>

<!--
```